Accurate and Efficient Curve Detection in Images: The Importance Sampling Hough Transform *

Daniel Walsh
Pennsylvania State University

Adrian E. Raftery
University of Washington

Technical Report No. 388
Department of Statistics
University of Washington
February 2001

*Daniel Walsh is Postdoctoral Research Associate, Department of Anthropology, Pennsylvania State University, University Park, PA 16802. Corresponding author e-mail: dcw11@psu.edu. Adrian E. Raftery is Professor of Statistics and Sociology, Department of Statistics, University of Washington, Box 354322, Seattle, WA 98915-4322. E-mail: raftery@stat.washington.edu; Web: www.stat.washington.edu/raftery.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **FEB 2001**	2. REPORT TYPE	3. DATES COVERED **00-02-2001 to 00-02-2001**

4. TITLE AND SUBTITLE **Accurate and Efficient Curve Detection in Images: The Importance Sampling Hough Transform**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Washington,Department of Statistics,Box 354322,Seattle,WA,98195-4322**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**24**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Abstract

The Hough transform is a well known technique for detecting parametric curves in images. We place a particular group of Hough transforms, the probabilistic Hough transforms, in the framework of importance sampling. This framework suggests a way in which probabilistic Hough transforms can be improved: by specifying a target distribution and weighting the sampled parameters accordingly to make identification of curves easier. We investigate the use of clustering techniques to simultaneously identify multiple curves in an image. We also use probabilistic arguments to develop stopping conditions for the algorithm. The resulting methodology is called the Importance Sampling Hough Transform (ISHT). We apply our method to both simulated and real data, and compare its performance with that of two much used versions of the Hough transform: the standard Hough transform and the randomized Hough transform. In our experiments, it is more accurate than either of these common methods, and it is faster than the randomized Hough transform.

KEY WORDS: Clustering, Importance sampling, Hough transform, Probabilistic Hough transform, Target distribution.

Contents

1 Introduction **4**

1.1 Notation and Definitions 4

1.2 The Standard Hough transform 4

1.3 The Randomized Hough transform 5

2 The Importance Sampling Hough Transform **7**

2.1 Importance Sampling 7

2.2 The Algorithm 8

2.3 Importance Sampling Distribution 9

2.4 Target Distribution 9

2.5 Peak Detection in the Parameter Space via Clustering 10

2.6 Stopping Conditions 11

3 Examples **13**

3.1 Simulated Data 13

3.1.1 Implementation 14

3.1.2 Results 14

3.1.3 Stopping Conditions 16

3.2 Blood Cell Data 17

3.2.1 Implementation 17

3.2.2 Results 19

4 Discussion **21**

1 Introduction

The Hough transform is a well known technique for detecting parametric curves in images. We place a particular group of Hough transforms, the probabilistic Hough transforms, in the framework of importance sampling. This framework suggests a way in which probabilistic Hough transforms can be improved: by specifying a target distribution and weighting the sampled parameters accordingly to make identification of curves easier. We investigate the use of clustering techniques to simultaneously identify multiple curves in the image. We also use probabilistic arguments to develop stopping conditions for the algorithm. Results from applying our method and two popular versions of the Hough transform to both simulated and real data are shown.

1.1 Notation and Definitions

The Hough transform (Hough 1962), hereafter HT, is typically used to detect object boundaries in images. Before applying the HT to a particular image, the image must be transformed by an edge detection algorithm into a binary edge detected image (also referred to as the *image space*). An edge detection algorithm assigns each pixel to be a *foreground pixel* (*edge pixel*), or a *background pixel.* In all the examples presented here the foreground pixels are black and the background pixels are white. We will use the term *points* to refer to foreground pixels and the term *pixels* to refer to foreground and background pixels. We define N to be the total number of points in the image space.

In order to detect instances of a particular parametric curve in an image, one must decide upon a parameterization of the curve. We will denote the curve parameterization by Θ. The dimension of the curve will be denoted by p. For instance, if a line is parameterized in Cartesian coordinates, $p = 2$, and $\Theta = \{m, c\}$, where m and c are the slope and intercept of the line, respectively.

Several of the HTs we discuss use an *accumulator array* to detect curves. An accumulator array, denoted by $\mathcal{A}$, is a discretization of the parameter space, Θ. Each cell of $\mathcal{A}$ is used to store votes for curves whose parameters are contained within the cell.

1.2 The Standard Hough transform

The standard HT (SHT) is a popular method for detecting parametric curves (lines, circles, ellipses etc) in binary image data. Good introductions to the HT and surveys of the literature can be found in Illingworth and Kittler (1988) and Leavers (1993). We will briefly describe the SHT for line detection. Polar coordinates ($p = x\cos\theta + y\sin\theta$), are most often used to parameterize lines (Duda and Hart 1972). Given this parameterization, the parameters of all lines passing through a particular point (x', y') in the image space form a sinusoid in the parameter space. The standard HT for detecting lines proceeds by mapping each point in the image space to its sinusoid in the discretized parameter space (the accumulator array, $\mathcal{A}$). Each cell in the accumulator array is incremented once for each sinusoid that passes through it. A peak detection method, sometimes a simple threshold, is used to locate local peaks in the accumulator array. The location of each peak gives the parameters of each detected line.

1.3 The Randomized Hough transform

The main problems of the standard HT are its long computation times and large storage requirements. The long computation times are caused by the fact that the HT increments the cells in the accumulator array corresponding to *all* curves that pass through *all* points in the image. Thus, much of the computation time is spent storing votes for curves with very little support from the data. The storage requirements of the standard Hough transform are a bigger problem though, since the size of the accumulator array is exponential in the number of parameters. When the number of parameters is greater than two, the storage space requirements become excessive.

A new class of HTs called *probabilistic Hough transforms* (PHTs) attempted to address these problems by using *random sampling* of edge points, and *many-to-one mapping* of edge points from the image space to the parameter space. The simplest PHT is the randomized HT (RHT) due to Xu, Oja, and Kultanen (1990). Below we outline the RHT algorithm for the general case of detecting all instances of a particular p-dimensional curve in a binary image.

1. Create the set E of all edge points in the binary edge detected image.
2. Select p points, $(e_1, \ldots, e_p)$, at random from E.
3. Solve for the parameters (θ) of the curve in the image space, defined by the selected points.
4. Increment the appropriate cell, $\mathcal{A}(\theta)$, in an accumulator array.
5. If the $\mathcal{A}(\theta)$ exceeds a predefined threshold t, then the curve parameterized by θ is detected. When this happens, the points that lie on the curve are removed from E and the accumulator array is re-initialized.
6. Regardless of whether a curve is detected, check whether or not a stopping condition is satisfied. If it is not, return to step 2.

The two main ingredients in this algorithm (and all PHTs) are a *sampling mechanism* (steps 2-3) and a *peak detection method* (steps 4-5).

The sampling mechanism defines a sampling distribution on the continuous parameter space, Θ. However, the sampling mechanism is discrete in nature, so we can view the sampling mechanism as defining a discrete sample space $\Omega \subset \Theta$, and a sampling distribution (probability mass function), $g(\cdot)$, on Ω. In the RHT, the sampling mechanism is the selection of p points at random. Thus, if there are $K = \binom{N}{p}$ p-tuples of points and θ_i is the parameter for the i^{th} p-tuple, then $\Omega = \{\theta_1, \ldots, \theta_K\}$ and the sampling distribution can be written as:

$$g(\theta^*) = \frac{1}{K} \sum_{i=1}^{K} 1_{\{\theta^* = \theta_i\}} \qquad \theta^* \in \Omega.$$

If all the $\theta_i's$ are unique, then $g(\cdot)$ is a uniform distribution on Ω, i.e.

$$g(\theta^*) = \frac{1}{K} \qquad \theta^* \in \Omega.$$

Curves that are present in the image should correspond to regions in Θ that have relatively large probability under g. The peak detection method of the RHT achieves this by grouping sampled parameter values into cells in the accumulator array and comparing with a threshold.

Applying the RHT is not straightforward if the curve is nonlinear with respect to its parameters, in which case selecting p points does not necessarily uniquely define a p-dimensional curve. An ellipse is an example of a curve that is nonlinear with respect to its parameters. A solution to the problem of detecting ellipses within the framework of PHTs has been explored by McLaughlin (1998) (based on work by Yuen, Illingworth, and Kittler (1989) in a non-PHT framework). In his method, three points are selected at a time. Estimates of the tangents to the (possible) ellipse at these points are then calculated and used to interpolate the center of the ellipse. Given the location of the ellipse center the remaining parameters are easily determined.

A greater problem affecting the RHT is that its performance is poor when the image is noisy or complex. New PHTs have been proposed to improve the performance of the RHT. The simplest modification to the RHT is the RHT with point distance criterion (RHT_D). In this HT, at each iteration, one point is sampled at random and then $p-1$ points are sampled uniformly from all points that are within certain a distance (greater than d_{min} and less than d_{max}) of the first point. Sensible choices of d_{min} and d_{max} lead to an increase in the probability of sampling points on a curve present in the image.

Comparisons of various probabilistic (and non-probabilistic) HTs can be found in Kälviäinen, Hirvonen, Xu, and Oja (1995) and Kälviäinen and Hirvonen (1997). These methods include the RHT_D mentioned above, the Dynamic RHT, the Random Window RHT, the Window RHT, the Connective RHT, and the Dynamic Combinatorial HT. Most of these modifications to the RHT attempt to improve the sampling distribution in various ways so that it is more peaked around the parameters corresponding to the curves in the image, thus making the subsequent process of peak detection easier. In some cases the distinction between the sampling mechanism and peak detection method is blurred, e.g. in the DRHT where curves are detected by an iterative process involving two RHTs.

While all of these methods differ in the sampling distribution employed they all share two common traits: (i) they all detect curves sequentially, and (ii) curves are detected when the number of votes in a cell in the accumulator array exceeds a certain threshold, or in other words, when a curve has been sampled a certain number of times. We see these as areas where improvements can be made. When detecting curves sequentially, every time a curve is detected the accumulator array is reset, thus discarding other curves that have already been accumulated. When accumulating a curve in the accumulator array, typically no effort is made to assess the "quality" of the entire curve. A practical consequence of this is that a curve must be sampled many times to be detected.

In our approach we will use a criterion for judging the "quality" of a curve, and introduce a technique that allows detection of multiple curves simultaneously. The form of this criterion is suggested by considering the following question: "What distribution would I *ideally* wish to sample from?" If we can *define* an ideal sampling distribution and obtain a sample from $g(\cdot)$, then we can obtain a sample from the ideal distribution via the technique of importance sampling.

In the following sections we introduce importance sampling and then show how it relates to the HT. We present some simple rules-of-thumb for determining how many samples from the sampling distribution are needed and suggest using clustering techniques to simultaneously identify curves. Results from applying these ideas to both simulated and real data are shown, and our method is compared to two standard HTs.

2 The Importance Sampling Hough Transform

2.1 Importance Sampling

Importance sampling is a technique that can be useful when a sample from a particular target distribution is desired but simulation from that distribution is not straightforward. We will restrict our attention to the case where the target distribution is discrete and known only up to a multiplicative scale factor.

Consider a discrete random variable, θ, with probability mass function $\{\pi(\theta) : \theta \in \Omega\}$ where Ω is the sample space of θ. We wish to obtain a sample $\theta_1, \ldots, \theta_T$ from $\pi(\cdot)$. We know the form of $\pi(\cdot)$ up to a scale factor, i.e.:

$$\pi(\theta) = f(\theta)/c, \quad \theta \in \Omega,$$

where the form of $f(\cdot)$ is known but the normalizing constant c is unknown. We now assume that there exists a probability mass function $g(\cdot)$ (the *sampling distribution* or *importance sampling function*) defined on Ω from which we can draw a sample, $\theta_1, \ldots, \theta_T$. Each observation in this sample is then weighted as follows:

$$w_i = \frac{f(\theta_i)/g(\theta_i)}{\sum_{j=1}^{T}\{f(\theta_j)/g(\theta_j)\}}.$$

These weights are called the *importance weights.* If a random (unweighted) sample from the target distribution is required (e.g. for plotting purposes), then *sampling/importance resampling* (SIR) (Rubin 1987) can be used. This consists of simply resampling the sampled parameters, with replacement, with probabilities proportional to their importance weights.

2.2 The Algorithm

We now outline an algorithm for a new PHT that can be viewed in an importance sampling framework. We call it the Importance Sampling Hough Transform (ISHT). The algorithm to detect curves, parameterized by Θ, in a binary edge detected image proceeds as follows.

1. Create the set E of all edge points in the binary edge detected image.
2. Define a target distribution $f(\theta|E)$. This is a function that measures the "quality" or "goodness-of-fit" of the curve given by θ, to the edge points E. For convenience we shall write $f(\theta)$.
3. Obtain a random sample, or *batch*, of parameters, $\{\theta_1, \ldots, \theta_{T^*}\}$, of size T^* from a sampling distribution $g(\cdot)$.
4. For each sampled parameter, θ_i, in the batch, calculate its importance weight as:

   $$w_i = \frac{f(\theta_i)}{\sum_{j=1}^{T^*} f(\theta_j)}.$$

   (For simplicity we have assumed that $g(\cdot)$ is approximately uniform over its range and thus does not appear in the above equation. See Sections 2.3 for discussion of the implications of this assumption.)
5. Run a peak detection method on the weighted sample of parameters to identify the curve parameters. Remove the points corresponding to each curve identified from E.
6. Check whether or not certain stopping conditions have been satisfied. If not return to step 3.

The four main components of this algorithm are: a sampling distribution, a target distribution, a peak detection method, and suitable stopping conditions. Each of these components is discussed below in greater detail.

2.3 Importance Sampling Distribution

Once a sampling mechanism is chosen it defines the importance sampling distribution, $g(\cdot)$. As with all PHTs, the sampling mechanism is vital to an efficient HT. If an image is moderately complex (e.g. several curves present) then the simplest sampling mechanism (sampling p points at random from the entire image) will rarely sample curve parameters. The easiest modification to this sampling mechanism is to use the point distance criterion (see Section 1.3). However, even though this sampling mechanism is easy to implement, exact calculation of $g(\cdot)$ is difficult. This is also true of more complex sampling mechanisms. In these cases we make the simplifying assumption that the sampling distribution is approximately uniform and therefore cancels from the calculation of the importance weights (i.e $w_i \propto f(\theta_i)$). This assumption has little effect on the performance of the Hough transform.

2.4 Target Distribution

The selection of a good target distribution is crucial to the success of the HT. A good target distribution will have a large concentration of its mass on the parameters corresponding to curves present in the image. The target distributions we consider are of the form:

$$f(\theta_i) = \beta(\theta_i) \times \alpha(\theta_i),$$

where $\beta(\theta_i)$ is a measure of the number of points associated with the curve given by θ_i, and $\alpha(\theta_i)$ is a measure of the goodness of fit of these points to the curve.

We considered several functions for $\beta(\cdot)$.

1. $\beta(\theta_i) = n_i \times 1_{\{n_i > t_n\}}$, where n_i is the number of points that lie on the curve given by θ_i, and t_n is a predefined threshold. If $\alpha(\cdot) \equiv 1$, then this can be thought of as the target distribution corresponding to the SHT.
2. $\beta(\theta_i) = \sum_{j=1}^{N} W(r_j)$, where r_j is the minimum distance from the j^{th} point to the curve given by θ_i, and $W(r_j)$ is a weighting function.

This weighting function can be interpreted as a fuzzy membership function with *membership radius* R if $W(0) = 1$, $W(r)$ decreases monotonically from 0 to R, and $W(r) = 0$ for $r \geq R$ (Han, Koczy, and Poston 1994). Natural choices for $W(r)$ are:

1. Step-function:
   $$W(r) = \begin{cases} 1, & 0 \leq r < R \\ 0, & \text{else} \end{cases}$$

2. Gaussian:
$$W(r) = \begin{cases} e^{-r^2/\sigma^2}, & 0 \leq r < R \\ 0, & \text{else} \end{cases}$$

3. Linear:
$$W(r) = \begin{cases} 1 - r/R, & 0 \leq r < R \\ 0, & \text{else} \end{cases}$$

Note that for values of R between $\frac{1}{2}$ and $\frac{1}{\sqrt{2}}$, if one uses the Step weighting function then $\beta(\theta_i) = \sum_{j=1}^{N} W(r_j) \approx n_i$, the number of points that lie on the curve .

The various goodness-of-fit criteria we considered were:

1. $\alpha(\theta_i) = (n_i/c_i)^G$, where n_i is defined as above and where c_i is the number of *pixels* that lie on the curve. G is a positive number. For $G = 1$, this is the proportion of pixels on the curve that are points.

2. $\alpha(\theta_i) = (p_i/c_i)^G$. Each point within distance membership radius R of the curve is projected onto the curve. The number of these projected points is p_i and c_i is defined as above.

3. $\alpha(\theta_i) = (p_i/c_i)^G \times 1_{\{(p_i/c_i)^G > t_\alpha\}}$, for some threshold t_α.

All of the above criteria lie between 0 and 1. If $G > 1$ then it can be thought of as a penalty term, penalizing "bad" curves. However if $G < 1$ then the goodness-of-fit of each curve is boosted. The first goodness-of-fit criterion listed is very strict. It favors only "perfect curves" whereas the second criterion allows variation around the curve. The last criterion gives zero weight to curves that do not exceed a certain goodness-of-fit threshold.

Obviously there is a great variety of target distributions that can be used. The application will play a large role in determining which target distributions are suitable.

2.5 Peak Detection in the Parameter Space via Clustering

Given a sample of parameters from the sampling distribution, and their importance weights, a peak detection method must be used to determine the number of curves, and their parameters. This simultaneous detection of peaks in the parameter space can be thought of as a clustering problem. We wish to partition the sampled parameters into spatially compact groups, where each group represents possible parameters for a particular curve in the image. The center of each group will be our estimate of the parameters for that curve. Of course some sampled parameters do not correspond to any curve in the image. These parameters can be thought of as noise. The number of these sorts of parameters in the sample will depend on the efficiency of the sampling distribution. However, these parameters should all have low importance weights. They can be removed by thresholding the sample based on the importance weights. Once these noise parameters are removed, the remaining sample of parameters can be clustered using any one of numerous clustering methods. These range from the simple k-means clustering (Hartigan 1975) (for a range of values of k) to more complex hierarchical and model-based methods, such as MCLUST (Banfield and Raftery 1993; Fraley and Raftery 1998).

We will use a simple method similar in spirit to the peak detection method used in the RHT. First, we threshold the sampled parameters based on their importance weights. Each of the remaining parameters are assumed to be associated with a curve in the image. We then place each parameter in a cell of an accumulator array (as in the RHT) and run a connected components algorithm on the non-empty cells in the array. The parameters contained in the cells associated with a particular connected component are assumed to be associated with one curve in the image. We estimate the parameters of this curve by taking a weighted average of all the parameters associated with the component (where the weights used are the importance weights, w_i). Choosing the cell size appropriately is necessary for the curves to be detected accurately. This requirement is common to all PHTs proposed to date, though.

2.6 Stopping Conditions

One advantage of our approach over other PHTs is that the we do not have to sample the points that lie on a curve many times in order to detect the curve. This is because our algorithm incorporates a measure of the quality of each curve. As a result of this, simple probabilistic arguments can be used to develop stopping conditions for the HT.

Our notation will be as follows. Let b be the number of batches currently processed, $T = bT^*$ be the total number of parameters sampled, and b_0 be the current number of consecutive batches processed without detecting a single curve. Let the current number of points in the edge set, E, be N_b.

We propose two different stopping conditions to determine if a sufficient number of parameters have been sampled from $g(\cdot)$. One condition is to stop the HT if the total number of parameters sampled, T, exceeds some threshold, T_{max}. An alternative rule is to stop the HT if the number of consecutive batches processed without detecting a single curve, b_0, is greater than some threshold, b_0^{max}. Of course if the number of remaining edge points becomes zero (or undesirably low) then sampling should stop.

Stopping Condition 1: Total number of parameters sampled

The total number of parameters sampled, T, will be sufficient if the probability of sampling all curves present in the image at least once is high. We can write down this probability if we have some idea of the number and size of the curves in the image. For example consider an image containing M p-dimensional curves of n points each. Let $p_{\{n,N\}}$ be the probability of sampling p points all from a n point curve, given that the image contains a total of N points. The probability of having sampled all curves after T samples is:

$$Pr = 1 - \sum_{i=1}^{M} (-1)^{i-1} \binom{M}{i} \left(1 - i \cdot p_{\{n,N\}} \right)^{T}.$$

This probability can be plotted as a function of T and can then be used to gauge what would constitute a reasonable sample. Derivation of this probability can be found in Appendix 4.

If the sampling mechanism is simply sampling p points at random then

$$p_{\{n,N\}} = \frac{\binom{n}{p}}{\binom{N}{p}}.$$

If one uses the point distance criterion then this probability can be approximated by

$$p_{\{n,N\}} \approx \frac{n}{N} \times \frac{\binom{n'-1}{p-1}}{\binom{N'-1}{p-1}}.$$

Here n' is the expected number of points that would typically lie on a given curve that would satisfy the point distance criterion given that the first point selected lies on that given curve. Similarly N' is the total number of points that typically satisfy the point distance criterion given selection of the first point. This condition will result in a conservative stopping condition (provided the guesses for M, n, and N are good). This is because the above calculation does not take into account the fact that, as curves are detected and removed from the image, the probability of detecting the remaining curves increases.

Stopping Condition 2: Number of batches processed since last detection

Consider the number of consecutive batches processed without detecting a single curve, b_0. If this number is not zero, then we should cease sampling if the probability of not having sampled a curve over these $b_0 T^\star$ samples, given that one remains, is low. Call this probability δ. This can be calculated as follows:

$$\delta = (1 - p_{\{n,N_b\}})^{T}.$$

Sampling should stop if δ is low enough (less than a predefined threshold δ_0, say), or alternatively if:

$$b_0^{max} = \frac{1}{T^*} \frac{\log \delta_0}{\log(1 - p_{\{n,N_b\}})} \leq b_0.$$

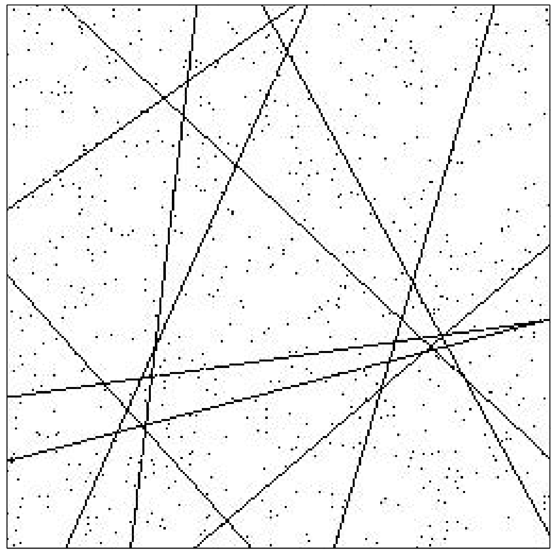


Figure 1: Simulated 256×256 binary image containing 10 lines and 1% speckle noise.

This stopping condition is preferable to the first one, since it uses more information about the current state of the HT (e.g. how many curves have been detected, which edge points remain) and should result in a smaller sample size being needed. Discussion of the selection of the batch size can be found in Section 4.

3 Examples

In this section we show some results obtained by applying our methods to both real and simulated data. We compare our method (ISHT) to the popular randomized Hough transform with point distance criterion (RHT_D), and to the original standard Hough transform (SHT).

3.1 Simulated Data

We simulated a 256×256 binary image, containing 10 lines (see Figure 1). In this image we randomly changed the color of each pixel with probability 0.01. The total number of points in the image is 2809, while the total number of points associated with lines is 2192. However, the probability of sampling a pair of points at random both from the same line is only about 0.08.

3.1.1 Implementation

For the ISHT, we chose the sampling distribution corresponding to the sampling mechanism of the RHT with point distance criterion with $d_{min} = 1$, and $d_{max} = 50$, i.e. the two pixels selected must be within 50 pixels of each other. The components of our target distribution were given by:

- $\beta(\theta_i) = \sum_{j=1}^{N} W(r_j)$, where $W(\cdot)$ is the Gaussian weighting function with $R = 2$, and $\sigma = 2$.
- $\alpha(\theta_i) = (p_i/c_i) \times 1_{\{(p_i/c_i)>0.5\}}$.

This target distribution allows points within 2 pixels of the line to be associated with the curve, but penalizes curves that do not have at least half as many points near the line as pixels on the line. We used a peak detection method with an importance weight threshold of zero. The cell size of the accumulator array (in $\{\rho, \theta\}$ notation) was $1.8 \times \frac{2\pi}{100}$. The batch size was 50, and the HT was stopped if two consecutive batches failed to detect any curves.

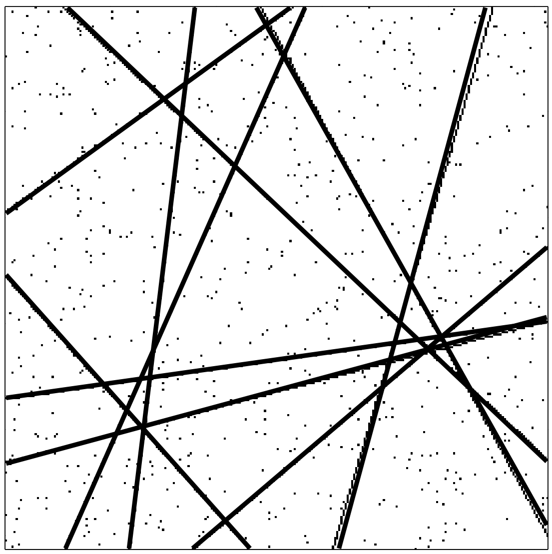
For the RHT_D, we set the threshold to be 10 points, and used the same point distance criterion as the ISHT above. We ceased sampling when the number of points remaining in the image was less than 30% of the original total. The cell size of the accumulator array was $1.8 \times \frac{2\pi}{100}$. When removing detected curves from the image we removed all points within a 3 pixel radius of the curves.

For the SHT, the threshold was set at 130 points. The cell size of the accumulator array was $3.6 \times \frac{\pi}{100}$. After thresholding the accumulator array (all cells below the threshold were set to zero), we ran a connected components algorithm on the non-empty cells. The cell with the most votes inside each connected component was identified as corresponding to a curve present in the image. The parameters of the curve were given by the parameters of the center of the cell. This curve identification procedure is similar to our ISHT clustering algorithm.

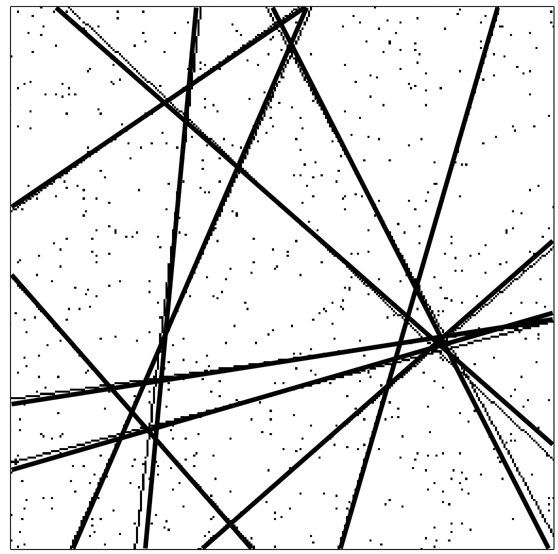
3.1.2 Results

The lines detected by each method are shown in Figure 2. Each method detected all 10 lines in the image with no false positives, but the three methods differed substantially in speed and accuracy. Table 1 shows the run times of each method, and the mean squared errors for ρ and θ.

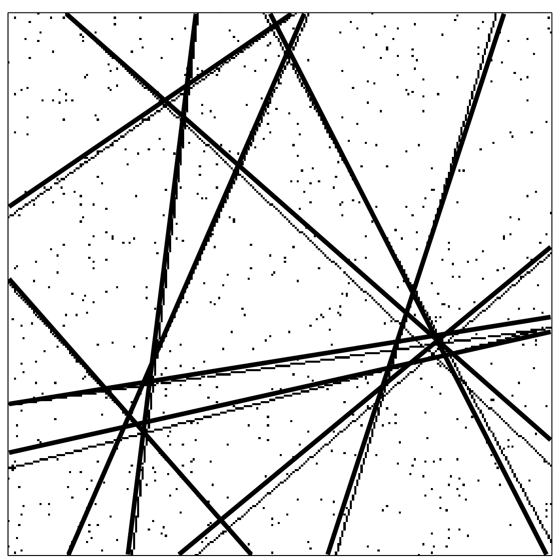
In terms of accuracy, the ISHT was clearly the best. The MSE for ρ was nearly an order of magnitude better for the ISHT than for the RHT_D, and two orders of magnitude better than for the SHT. The MSE for θ for the ISHT was 4 times lower than for either of the older methods. The SHT was the fastest method, approximately 3 times faster than the ISHT, which was an order of magnitude faster than the RHT_D. This is not too surprising given



(a) ISHT.



(b) RHT_D.



(c) SHT.

Figure 2: Simulated data : The thin lines are the true lines, and the thick lines are the lines detected by each algorithm.

Table 1: Simulated Data: Run times, and mean squared errors for each HT.

	ISHT	RHT_D	SHT
Run Time (seconds)	4.73	44.64	1.770
MSE of ρ	0.061	0.362	5.293
MSE of θ ($\times 10^5$)	9.02	40.91	45.65

that the parameter space was of low dimension (high dimension adversely affects the SHT more than the RHT), and that the image was complex (the complexity of the image affects the RHT more than the SHT).

3.1.3 Stopping Conditions

For this example we used the stopping condition based on the number of consecutive empty batches. Specifically, we stopped the ISHT when we had two empty batches, each of size 50 parameters. The final image, after the detected lines had been removed, contained 692 points. It would be interesting to calculate the probability of not detecting a line, given that one existed, in these remaining 100 (50×2) samples. Suppose, for simplicity, that our sampling mechanism, had been that of the RHT, and the undetected line contained 100 points. In this case, the probability of sampling two points on this line is:

$$\begin{aligned} p_{\{n,N_b\}} &= \binom{n}{2} \Big/ \binom{N}{2} \\ &\approx (n/N)^2 \\ &= (100/692)^2 \\ &= 0.0209. \end{aligned}$$

Therefore the probability of not detecting the curve is:

$$\begin{aligned} Pr &= (1 - p_{\{n,N_b\}})^{100} \\ &= (1 - 0.0209)^{100} \\ &= 0.12. \end{aligned}$$

This probability is low, although not very low. It suggests that maybe we should run our HT longer. The alternative stopping condition, based on the total number of parameters sampled, is more conservative. Figure 3 shows the probability of sampling all curves in the image for a given total sample size, assuming there are either 10 or 20 lines in the

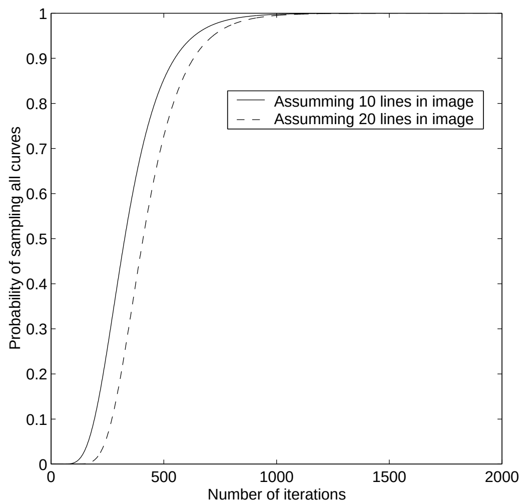


Figure 3: Simulated Data: Probability of sampling all lines in the image by a given iteration, assuming there are 10 or 20 lines in the image.

image (where each line is of size 256 points). The probability for both curves is high when the number of parameters sampled is approximately 750. In this example we sampled 250 parameters in total, and we can see that the probability is only approximately 0.5. Since we had detected all the curves in the image after 250 samples, it appears that this stopping condition can be too conservative; however, it is a good rule of thumb for establishing an upper bound for the number of parameters sampled.

3.2 Blood Cell Data

Figure 4 shows a 265×272 image of blood cells. The edge detected image, obtained via a Sobel transform, is shown in Figure 5. The task of identifying each blood cell is not straightforward. There are numerous blood cells in the image. All are roughly circular, although not perfectly so, and several are occluded, either by other blood cells or by the boundary of the image.

3.2.1 Implementation

We used the same circle parameterization for each HT, namely $\Theta = \{r, a, b\}$, where r is radius of the circle, and $\{a, b\}$ is the row and column position of the circle center (all in pixels).

For the ISHT, we chose a batch size of 200 samples. The sampling mechanism and target distribution are the same as in the previous example. Again we clustered the sampled

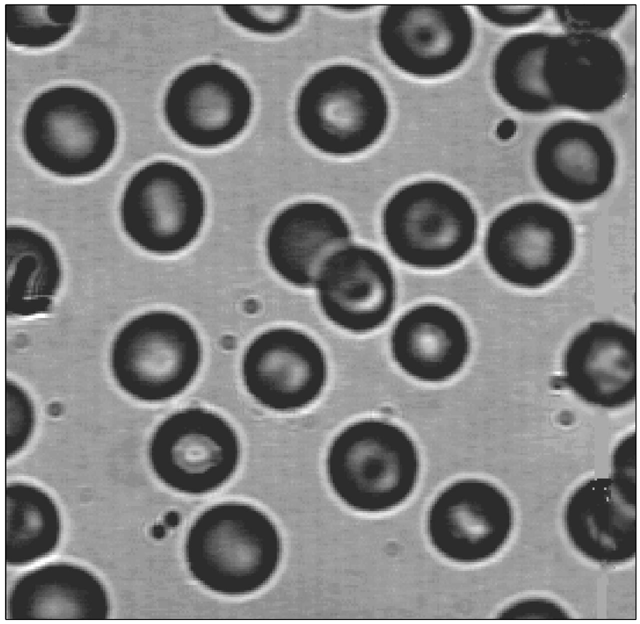


Figure 4: Light microscope image of red blood cells. Original image from *The Image Processing Handbook*, (Russ 1999). Used with permission.

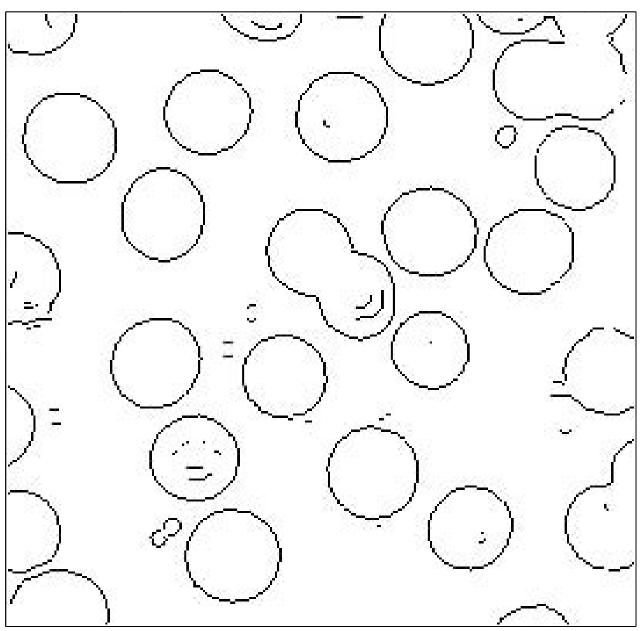


Figure 5: Blood cells - Edge detected image.

parameters after each batch using our peak detection method. The cell size used was (in $\{r, a, b\}$ notation) $0.8 \times 11 \times 11$ and the sample was thresholded based on an importance weight of 45. The ISHT was stopped if 4 consecutive batches detected no curves.

For the RHT_D, we set the threshold of the RHT_D to be 10 points, and used the same point distance criterion as the ISHT above. The size of a cell in the accumulator array was $0.8 \times 5 \times 5$. As in the previous example, we stopped sampling when the number of points remaining in the image was less than 30% of the original total, and we used a 3 pixel radius when determining which points should be removed after a curve had been detected.

In order to alleviate the severe storage requirements of the SHT once the number of parameters reaches 3, as here with circles, we employed the Gerig-Klein modification to the HT (Gerig and Klein 1986). This modified HT allows only one circle to be detected at each given center. The reduces the accumulator array from a three-dimensional array to two two-dimensional arrays. The cost is the inability to detect concentric circles. Of course in this example that is a reasonable assumption.

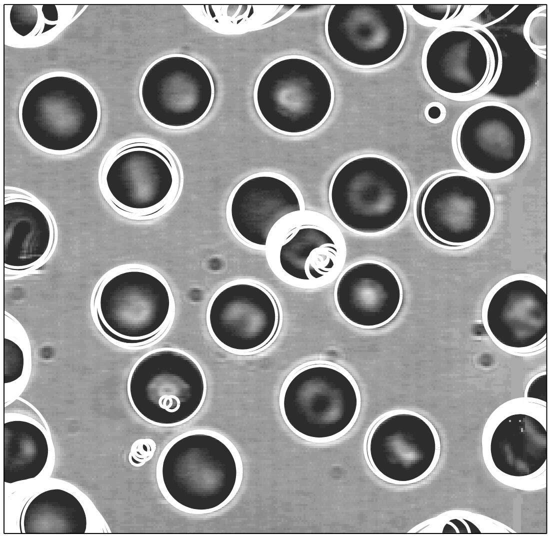
3.2.2 Results

The final classifications for each method are shown in Figure 6. We considered 26 of the blood cells in the image to be detectable. The number of cells correctly detected, along with the run times, the number of false positives, and the number of duplicates are shown in Table 2.

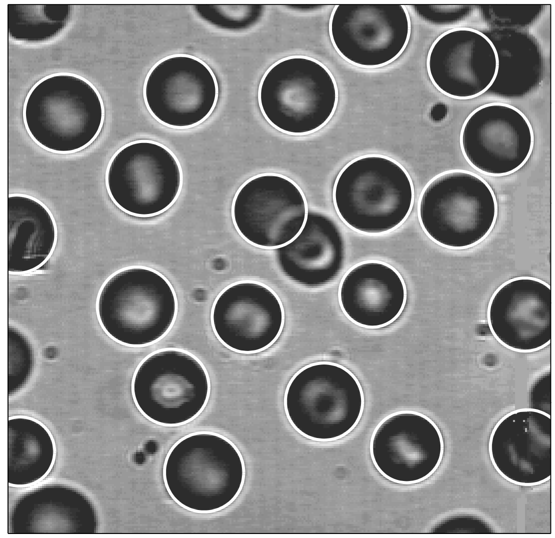
Table 2: Blood Cell Analysis.

	ISHT	RHT_D	SHT
Run time	44.90	263.6	138.9
Cells detected	21	23	20
Cells undetected	5	3	6
False positives	0	13	0
Duplicates	0	2	5

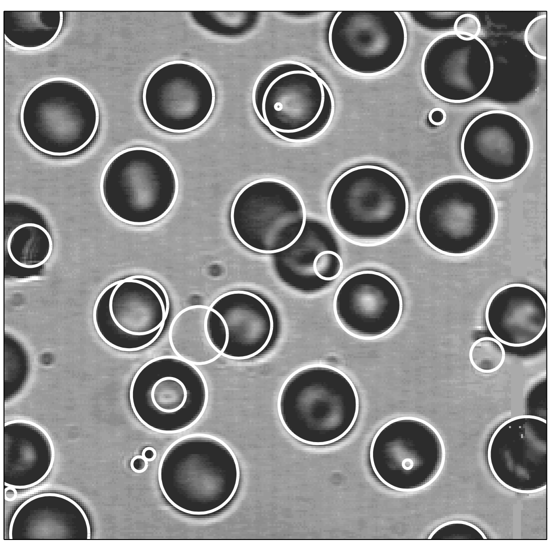
The ISHT detected 21 of the 26 blood cells, detected no false positives and no duplicates, and had by far the shortest run time. The RHT_D detected more blood cells (23), but also detected 13 false positives. The SHT detected fewer blood cells (20), and also detected 5 duplicates. We can see by visually comparing the images in Figure 6 that the circles detected by the ISHT fit the blood cells better. The ISHT did not detect several of the cells that are not fully within the image, as well as the two partially occluded cells that are fully contained within the image. However, with suitable modifications of the target distribution these cells could conceivably be detected.



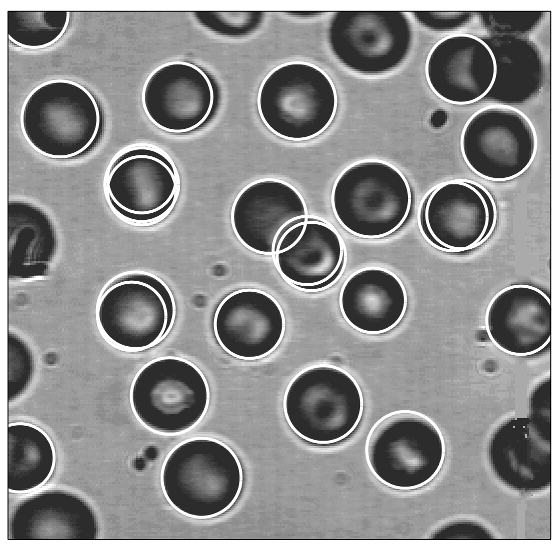
(a) ISHT - All circles corresponding to sampled parameters with positive importance weight.



(b) ISHT - Circles detected after clustering.



(c) RHT_D - Circles detected.



(d) SHT - Circles detected.

Figure 6: Blood Cells: Circles Detected.

4 Discussion

We have found importance sampling to be a natural framework with which to design, view, and understand probabilistic Hough transforms. We feel that the notion of a target distribution, which defines an importance weight associated with every sampled parameter, is fundamental to the successful implementation of a PHT. We investigated the feasibility of detecting multiple curves from a given batch of sampled parameters. We found that simple clustering methods provided good identification of curve parameters, provided that the sampled parameters not associated with curves present in the image were filtered out. The removal of these "noise" parameters was achieved via a simple thresholding of the importance weights. Probabilistic arguments can be used to determine reasonable stopping conditions for PHTs.

As with most data processing algorithms, their successful implementation often depends on correctly setting several tuning parameters. We feel that, depending on the given image, the tuning parameters of the ISHT are reasonably intuitive to set. One parameter, the batch size (if one is employing the second stopping condition) can affect the efficiency and accuracy of the ISHT. If the batch size is too large, the increased efficiency of sampling from a smaller pool of points is lost. Conversely, if the batch size is too small, curves may be detected after only being sampled once, thus increasing the bias in the parameter estimates. One possible way to avoid this problem is to use a small batch size, and employ an algorithm similar to the Dynamic RHT. That is, for each detected curve, select all the points near the curve and perform a refined HT on these points. This technique can be thought of as adaptively sampling the parameter space, and it is of course not the only conceivable approach to take.

The importance sampling framework is broad enough to incorporate many of the beneficial features of other existing PHTs. For example, we used a very simple sampling distribution, whereas one could use a more sophisticated sampling distribution, e.g. one based on the Connective RHT. In addition, the identification of curves could be improved by incorporating an adaptive sampling mechanism, such as in the Dynamic RHT. The main computational burden of our approach over other PHTs is the evaluation of the target distribution. This should be an $O(N)$ operation, where N is the total number of points in the image. We feel that the benefit of evaluating the quality of a curve far outweighs the cost of its computation. In summary, we feel that importance sampling provides a framework in which different PHTs can be understood, and which will be useful in guiding the design of better curve detection methods.

Acknowledgments

This research was supported by the Office of Naval Research grants N00014-97-1-0736 and N00014-96-1-0192.

Appendix A: Calculation of the Probability of Detecting All Curves

Given M events, $A_0, \ldots, A_{M-1}$, the probability of the union of all of these events can be expressed in the following way:

$$\begin{aligned}
P(\cup_{i=1}^{M} A_i) &= P(A_1) + \ldots + P(A_M) \\
&\quad +(-1)^1\{\sum_{i<j} P(A_i \cap A_j)\} \\
&\quad +(-1)^2\{\sum_{i<j<k} P(A_i \cap A_j \cap A_k)\} \\
&\quad \vdots \\
&\quad +(-1)^{M-1}\{P(\cap_{i=1}^{M} A_i)).\}
\end{aligned}$$

Now consider an image containing M p-dimensional curves of n points each. Assume a sampling distribution, $g(\cdot)$, exists that enables one to select p points from the N total points in the image.

Now consider a sampling scheme where, at each draw, the p points are sampled from the N total points using the sampling distribution. If, at a particular draw, all p points are sampled from the same curve, we shall consider that curve to have been sampled. Define $p_{\{n,N\}}$ to be the probability of sampling a curve at a particular draw (we assume for simplicity this is the same for all curves).

Let A_i^t be the event that the i^{th} curve has *not* been sampled after t draws of p points from the sampling distribution. Since each curve has n points and each sampled parameter is drawn independently from $g(\cdot)$, we can write:

$$P(A_i^t) = (1 - p_{\{n,N\}})^t, \text{ for } i = 1, \ldots, M.$$

Similarly $P(A_i^t \cap A_j^t)$ can be calculated as follows:

$$\begin{aligned}
P(A_i^t \cap A_j^t) &= (P(A_i^1 \cap A_j^1))^t \\
&= (1 - P((A_i^1 \cap A_j^1)^c))^t \\
&= (1 - P((A_i^1)^c \cup (A_j^1)^c))^t \\
&= (1 - (P((A_i^1)^c) + P((A_j^1)^c) - P((A_i^1)^c \cap (A_j^1)^c)))^t
\end{aligned}$$

$$
\begin{aligned}
&= (1 - p_{\{n,N\}} - p_{\{n,N\}} + 0)^t \\
&= (1 - 2 \cdot p_{\{n,N\}})^t, \text{ for all } i = 1, \ldots, M, j \neq i.
\end{aligned}
$$

The term $P((A_i^1)^c \cap (A_j^1)^c)$ is equal to zero since $(A_i^1)^c$ and $(A_j^1)^c)$ are mutually exclusive (one cannot sample two curves at the same time). Similarly, the probability of the intersection of all events can be written as:

$$P(\cap_{i=1}^M A_i^t) = (1 - M \cdot p_{\{n,N\}})^t.$$

Substituting these expressions into the first equation we find that the probability of *not* sampling all curves at least once after t samples have been drawn from $g(\cdot)$ is:

$$P(\cup_{i=1}^M A_i^t) = \sum_{i=1}^{M} (-1)^{i-1} \binom{M}{i} \left(1 - i \cdot p_{\{n,N\}} \right)^t.$$

The probability of having sampled all curves after t samples is simply this quantity subtracted from 1.

References

Banfield, J. D. and A. E. Raftery (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics 49*(3), 803–822.

Duda, R. O. and P. E. Hart (1972). Use of the Hough transform to detect lines and curves in pictures. *Communications of the Association for Computing Machinery : Graphics Image Processing 15*(1), 11–15.

Fraley, C. and A. E. Raftery (1998). How many clusters? Which clustering method? - Answers via model-based cluster analysis. *Computer Journal 41*(8), 578–588.

Gerig, G. and F. Klein (1986). Fast contour identification through efficient Hough transform and simplified interpretation strategy. In *Proceedings of the Eighth International Conference on Pattern Recognition*, pp. 498–500. Paris.

Han, J. H., L. T. Koczy, and T. Poston (1994). Fuzzy Hough transform. *Pattern Recognition Letters 15*(7), 649–658.

Hartigan, J. A. (1975). *Clustering algorithms.* New York: John Wiley & Sons. Wiley Series in Probability and Mathematical Statistics.

Hough, P. V. C. (1962). Method and means for recognizing complex patterns. U.S. Patent 3,069,654.

Illingworth, J. and J. Kittler (1988). A survey of the Hough transform. *Computer Vision Graphics and Image Processing 44*(1), 87–116.

Kälviäinen, H. and P. Hirvonen (1997). An extension to the randomized Hough transform exploiting connectivity. *Pattern Recognition Letters 18*(1), 77–85.

Kälviäinen, H., P. Hirvonen, L. Xu, and E. Oja (1995). Probabilistic and non-probabilistic Hough transforms - overview and comparisons. *Image and Vision Computing 13*(4), 239–252.

Leavers, V. (1993). Which Hough transform? A survey of Hough transform methods. *CVGIP - Image Understanding 58*(2), 250–264.

McLaughlin, R. A. (1998). Randomized Hough Transform: Improved ellipse detection with comparison. *Pattern Recognition Letters 19*(3-4), 299–305.

Rubin, D. B. (1987). Comment on 'The calculation of posterior distributions by data augmentation' by Tanner and Wong. *Journal of the American Statistical Association 82*(398), 543–546.

Russ, J. C. (1999). *The Image Processing Handbook* (Third ed.). Boca Raton, Florida: CRC press.

Xu, L., E. Oja, and P. Kultanen (1990). A new curve detection method: Randomized Hough Transform (RHT). *Pattern Recognition Letters 11*(5), 331–338.

Yuen, H. K., J. Illingworth, and J. Kittler (1989). Detecting partially occluded ellipses using the Hough transform. *Image and Vision Computing 8*(1), 71–77.